C 6

```
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
**FILE**ID**MWTUR1                                      K 2

MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR        11
MM      MM  WW      WW  TTTTTTTTTT  UU      UU  RRRRRRRR        11
MMMM  MMMM  WW      WW      TT      UU      UU  RR      RR    1111
MMMM  MMMM  WW      WW      TT      UU      UU  RR      RR    1111
MM  MM  MM  WW      WW      TT      UU      UU  RR      RR      11
MM  MM  MM  WW      WW      TT      UU      UU  RR      RR      11
MM      MM  WW      WW      TT      UU      UU  RRRRRRRR        11
MM      MM  WW      WW      TT      UU      UU  RRRRRRRR        11
MM      MM  WW  WW  WW      TT      UU      UU  RR  RR          11
MM      MM  WW  WW  WW      TT      UU      UU  RR  RR          11
MM      MM  WWWW  WWWW      TT      UU      UU  RR    RR        11          ....
MM      MM  WWWW  WWWW      TT      UU      UU  RR    RR        11          ....
MM      MM  WW      WW      TT      UUUUUUUUUU  RR      RR    111111        ....
MM      MM  WW      WW      TT      UUUUUUUUUU  RR      RR    111111        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE MWTUR1 (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MOUNT Utility Structure Level 1
1 !
1 ! ABSTRACT:
1 !
1 !       This module generates a window mapping the desired VBN from
1 !       the supplied file header. This module is a direct crib from the
1 !       FCP module WITURN. When we get conditional compilation with different
1 !       parameter files worked out, they should really be the same source.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein, CREATION DATE:  7-Dec-1976  14:38
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-001 HH0041          Hai Huang               24-Jul-1984
1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !       V02-000 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:39
```

```
:     58           0058   1 !                       Previous revision history moved to MOUNT.REV
:     59           0059   1 !**
:     60           0060   1
:     61           0061   1
:     62           0062   1 LIBRARY 'SYS$LIBRARY:LIB.L32';
:     63           0063   1 REQUIRE 'SRC$:MOUDEF.B32';
```

```
;    65          0595   1 GLOBAL ROUTINE TURN_WINDOW1 (WINDOW, HEADER, DESIRED_VBN, START_VBN) =
;    66          0596   1
;    67          0597   1 !++
;    68          0598   1 !
;    69          0599   1 ! FUNCTIONAL DESCRIPTION:
;    70          0600   1 !
;    71          0601   1 !       This routine scans the map area of the supplied file header
;    72          0602   1 !       and builds retrieval pointers in the window until
;    73          0603   1 !       (1) the entire header has been scanned, or
;    74          0604   1 !       (2) the first retrieval pointer in the window maps the desired VBN
;    75          0605   1 !
;    76          0606   1 ! CALLING SEQUENCE:
;    77          0607   1 !       TURN_WINDOW (ARG1, ARG2, ARG3, ARG4)
;    78          0608   1 !
;    79          0609   1 ! INPUT PARAMETERS:
;    80          0610   1 !       ARG1: address of window block
;    81          0611   1 !       ARG2: address of file header
;    82          0612   1 !       ARG3: desired VBN
;    83          0613   1 !       ARG4: starting VBN of file header
;    84          0614   1 !
;    85          0615   1 ! IMPLICIT INPUTS:
;    86          0616   1 !       NONE
;    87          0617   1 !
;    88          0618   1 ! OUTPUT PARAMETERS:
;    89          0619   1 !       updated window
;    90          0620   1 !
;    91          0621   1 ! IMPLICIT OUTPUTS:
;    92          0622   1 !       NONE
;    93          0623   1 !
;    94          0624   1 ! ROUTINE VALUE:
;    95          0625   1 !       1
;    96          0626   1 !
;    97          0627   1 ! SIDE EFFECTS:
;    98          0628   1 !       NONE
;    99          0629   1 !
;   100          0630   1 !--
;   101          0631   1
;   102          0632   2 BEGIN
;   103          0633   2
;   104          0634   2 MAP
;   105          0635   2       WINDOW          : REF BBLOCK,   ! pointer to window
;   106          0636   2       HEADER          : REF BBLOCK;   ! pointer to file header
;   107          0637   2
;   108          0638   2 LOCAL
;   109          0639   2       VBN,                            ! VBN in scanning window
;   110          0640   2       COUNT,                          ! retrieval pointer count
;   111          0641   2       COUNTER,                        ! loop counter
;   112          0642   2       LBN,                            ! retrieval pointer start LBN
;   113          0643   2       W_POINTER       : REF BBLOCK,   ! pointer to scan window
;   114          0644   2       M_POINTER       : REF BBLOCK,   ! pointer to scan header map area
;   115          0645   2       MAP_AREA        : REF BBLOCK;   ! pointer to start of header map area
;   116          0646   2
;   117          0647   2 MACRO
;   118          0648   2       WINDOW_MAP      = (.WINDOW+WCB$C_MAP)%; !start of window map area
;   119          0649   2
;   120          0650   2 ! Scan the window looking for the
;   121          0651   2 ! starting VBN of the header. If it is contained within the window, truncate
;  122        0652  2 ! the window so that it maps exactly up to the start of the header.
;  123        0653  2 ! If the starting VBN is not contained in the window, the entire window must be
;  124        0654  2 ! discarded. However, if the desired VBN precedes the header start VBN, we
;  125        0655  2 ! do nothing since the window is already best effort.
;  126        0656  2 !
;  127        0657  2
;  128        0658  2 W_POINTER = WINDOW_MAP;                              ! point to first retrieval pointer
;  129        0659  2 VBN = .WINDOW[WCB$L_STVBN];                          ! get starting VBN of window
;  130        0660  2
;  131        0661  2 IF
;  132        0662  3     BEGIN
;  133        0663  3     INCR J FROM 1 TO .WINDOW[WCB$W_NMAP]
;  134        0664  3     DO
;  135        0665  3         IF .VBN + .W_POINTER[WCB$W_COUNT] LSSU .START_VBN
;  136        0666  3         THEN
;  137        0667  4             BEGIN
;  138        0668  4             VBN = .VBN + .W_POINTER[WCB$W_COUNT];          ! accumulate running VBN
;  139        0669  4             W_POINTER = .W_POINTER + 6;          ! and move to next pointer
;  140        0670  4             END
;  141        0671  3         ELSE                                     ! VBN is in or precedes current pointer
;  142        0672  4             BEGIN
;  143        0673  4             IF .VBN LSSU .START_VBN
;  144        0674  4             THEN
;  145        0675  5                 BEGIN
;  146        0676  5                 W_POINTER[WCB$W_COUNT] = .START_VBN - .VBN;
;  147        0677  5                 W_POINTER = .W_POINTER + 6;      ! bump the pointer
;  148        0678  5                 WINDOW[WCB$W_NMAP] = .J;         ! truncate the window
;  149        0679  5                 EXITLOOP 0;
;  150        0680  5                 END
;  151        0681  4             ELSE
;  152        0682  4                 EXITLOOP -1;                     ! precedes the window
;  153        0683  4             END
;  154        0684  3     END
;  155        0685  3
;  156        0686  2 THEN                                                 ! header VBN is not in window
;  157        0687  3     BEGIN
;  158        0688  3     IF .DESIRED_VBN LSSU .START_VBN
;  159        0689  3     AND .START_VBN GTRU 1
;  160        0690  3     THEN
;  161        0691  3         RETURN 1                                 ! leave it alone
;  162        0692  3     ELSE
;  163        0693  4         BEGIN
;  164        0694  4         WINDOW[WCB$W_NMAP] = 0;                  ! flush the window
;  165        0695  4         WINDOW[WCB$L_STVBN] = .START_VBN;
;  166        0696  4         W_POINTER = WINDOW_MAP;                  ! point to first pointer
;  167        0697  3         END;
;  168        0698  2     END;
;  169        0699  2
;  170        0700  2 ! The window is now suitably initialized. Set up necessary pointers.
;  171        0701  2 ! Now scan the map area, extracting retrieval pointers.
;  172        0702  2 !
;  173        0703  2
;  174        0704  2 MAP_AREA = .HEADER + .HEADER[FH1$B_MPOFFSET]*2; ! point to map area
;  175        0705  2 M_POINTER = .MAP_AREA + FM1$C_POINTERS;          ! point to start of pointers
;  176        0706  2
;  177        0707  2 DECR COUNTER FROM .MAP_AREA[FM1$B_INUSE]/2 TO 1 DO
;  178        0708  3     BEGIN
```

```
MWTU
V04-

: S
: Ru
: El
: L
: Le
: Mo
: Co
```

```
;  179         0709 3
;  180         0710 3       COUNT = .M_POINTER[FM1$B_COUNT] + 1;                    ! get retrieval pointer count
;  181         0711 3       LBN = .M_POINTER[FM1$W_LOWLBN];                         ! low order LBN
;  182         0712 3       LBN<16,8> = .M_POINTER[FM1$B_HIGHLBN];                  ! and high order LBN
;  183         0713 3       M_POINTER = .M_POINTER + 4;                     ! update map pointer
;  184         0714 3
;  185         0715 3 ! See if this set of blocks is contiguous with the preceding (in the
;  186         0716 3 ! window). If so, merge them.
;  187         0717 3 !
;  188         0718 3
;  189         0719 3       IF .WINDOW[WCB$W_NMAP] NEQ 0          ! only if non-empty window
;  190         0720 3           AND .W_POINTER[WCB$W_PREVCOUNT] + .W_POINTER[WCB$L_PREVLBN] EQL .LBN
;  191         0721 3           AND .W_POINTER[WCB$W_PREVCOUNT] + .COUNT LSSU 65536
;  192         0722 3       THEN W_POINTER[WCB$W_PREVCOUNT] = .W_POINTER[WCB$W_PREVCOUNT] + .COUNT
;  193         0723 3
;  194         0724 3 ! Build a new retrieval pointer. If the window is full, shuffle the
;  195         0725 3 ! entries up by one. If this would cause the pointer mapping the
;  196         0726 3 ! desired VBN to fall off the top, we are done.
;  197         0727 3 !
;  198         0728 3
;  199         0729 3       ELSE
;  200         0730 4           BEGIN
;  201         0731 4           IF (.WINDOW[WCB$W_NMAP]+1)*6 + WCB$C_LENGTH
;  202         0732 4               GTRU .WINDOW[WCB$W_SIZE]
;  203         0733 4           THEN
;  204         0734 5               BEGIN
;  205         0735 5               IF .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT] GTRU .DESIRED_VBN
;  206         0736 5               THEN EXITLOOP;
;  207         0737 5
;  208         0738 5               WINDOW[WCB$W_NMAP] = .WINDOW[WCB$W_NMAP] - 1;
;  209         0739 5               WINDOW[WCB$L_STVBN] = .WINDOW[WCB$L_STVBN] + .WINDOW[WCB$W_P1_COUNT];
;  210         0740 5               CH$MOVE (.WINDOW[WCB$W_NMAP]*6, WINDOW_MAP+6, WINDOW_MAP);
;  211         0741 5               W_POINTER = .W_POINTER - 6;
;  212         0742 4               END;
;  213         0743 4
;  214         0744 4 ! Finally build the pointer and count it.
;  215         0745 4 !
;  216         0746 4
;  217         0747 4           W_POINTER[WCB$W_COUNT] = .COUNT;
;  218         0748 4           W_POINTER[WCB$L_LBN] = .LBN;
;  219         0749 4           W_POINTER = .W_POINTER + 6;
;  220         0750 4           WINDOW[WCB$W_NMAP] = .WINDOW[WCB$W_NMAP] + 1;
;  221         0751 3           END;
;  222         0752 3
;  223         0753 2       END;                                            ! end of header scan loop
;  224         0754 2
;  225         0755 2 RETURN 1;
;  226         0756 2
;  227         0757 1 END;                                                 ! end of routine TURN_WINDOW1


                                                                     .TITLE  MWTUR1
                                                                     .IDENT  \V04-000\


                                                                     .PSECT  $CODE$,NOWRT,2
```

**F

```
                                      0FFC 00000         .ENTRY   TURN_WINDOW1, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0595
                                                                  R10,R11                                     ;
                        5E         04   C2 00002         SUBL2    #4, SP                                      ;
                        58   04    AC   D0 00005         MOVL     WINDOW, R8                                  ; 0658
                             30    A8   9F 00009         PUSHAB   48(R8)                                      ;
                        56         6E   D0 0000C         MOVL     (SP), W_POINTER                             ;
                        50   2C    A8   D0 0000F         MOVL     44(R8), VBN                                 ; 0659
                        59   16    A8   9E 00013         MOVAB    22(R8), R9                                  ; 0663
                        54         69   3C 00017         MOVZWL   (R9), R4                                    ;
                        51   10    AC   D0 0001A         MOVL     START_VBN, R1                               ; 0665
                                   53   D4 0001E         CLRL     J                                           ;
                                   27   11 00020         BRB      3$                                          ;
                        52         66   3C 00022 1$:     MOVZWL   (W_POINTER), R2                             ;
                        52         50   C0 00025         ADDL2    VBN, R2                                     ;
                        51         52   D1 00028         CMPL     R2, R1                                      ;
                                   0B   1E 0002B         BGEQU    2$                                          ;
                        52         86   3C 0002D         MOVZWL   (W_POINTER)+, R2                            ; 0668
                        50         52   C0 00030         ADDL2    R2, VBN                                     ;
                        56         04   C0 00033         ADDL2    #4, W_POINTER                               ; 0669
                                   11   11 00036         BRB      3$                                          ; 0665
                        51         50   D1 00038 2$:     CMPL     VBN, R1                                     ; 0673
                                   10   1E 0003B         BGEQU    4$                                          ;
                86      51         50   A3 0003D         SUBW3    VBN, R1, (W_POINTER)+                       ; 0676
                        56         04   C0 00041         ADDL2    #4, W_POINTER                               ; 0677
                        69         53   B0 00044         MOVW     J, (R9)                                     ; 0678
                                   18   11 00047         BRB      6$                                          ; 0679
                D5      53         54   F3 00049 3$:     AOBLEQ   R4, J, 1$                                   ; 0665
                        51   0C    AC   D1 0004D 4$:     CMPL     DESIRED_VBN, R1                             ; 0688
                                   05   1E 00051         BGEQU    5$                                          ;
                        01         51   D1 00053         CMPL     R1, #1                                      ; 0689
                                   7A   1A 00056         BGTRU    9$                                          ;
                                   69   B4 00058 5$:     CLRW     (R9)                                        ; 0694
                   2C   A8         51   D0 0005A         MOVL     R1, 44(R8)                                  ; 0695
                        56         6E   D0 0005E         MOVL     (SP), W_POINTER                             ; 0696
                        51   08    AC   D0 00061 6$:     MOVL     HEADER, R1                                  ; 0704
                        50   01    A1   9A 00065         MOVZBL   1(R1), R0                                   ;
                        50      6140    3E 00069         MOVAW    (R1)[R0], MAP_AREA                          ;
                        57   0A    A0   9E 0006D         MOVAB    10(R0), M_POINTER                           ; 0705
                        51   08    A0   9A 00071         MOVZBL   8(MAP_AREA), R1                             ; 0707
           04   AE      51         02   C7 00075         DIVL3    #2, R1, 4(SP)                               ;
                             04    AE   D6 0007A         INCL     COUNTER                                     ;
                                   76   11 0007D         BRB      11$                                         ;
                        5B   01    A7   9A 0007F 7$:     MOVZBL   1(M_POINTER), COUNT                         ; 0710
                                   5B   D6 00083         INCL     COUNT                                       ;
                        5A   02    A7   3C 00085         MOVZWL   2(M_POINTER), LBN                           ; 0711
      5A        08      10         87   F0 00089         INSV     (M_POINTER)+, #16, #8, LBN                  ; 0712
                                   69   B5 0008E         TSTW     (R9)                                        ; 0719
                                   23   13 00090         BEQL     8$                                          ;
                        50   FA    A6   3C 00092         MOVZWL   -6(W_POINTER), R0                           ; 0720
                        50   FC    A6   C0 00096         ADDL2    -4(W_POINTER), R0                           ;
                        5A         50   D1 0009A         CMPL     R0, LBN                                     ;
                                   16   12 0009D         BNEQ     8$                                          ;
                        50   FA    A6   3C 0009F         MOVZWL   -6(W_POINTER), R0                           ; 0721
                        50         5B   C0 000A3         ADDL2    COUNT, R0                                   ;
            00010000    8F         50   D1 000A6         CMPL     R0, #65536                                  ;
                                   06   1E 000AD         BGEQU    8$                                          ;
```

```
                                    FA   A6               5B  A0 000AF        ADDW2   COUNT, -6(W_POINTER)                              ; 0722
                                                          40  11 000B3        BRB     11$                                               ;
                                         50               69  3C 000B5 8$:    MOVZWL  (R9), R0                                          ; 0731
                                         50               06  C4 000B8        MULL2   #6, R0                                            ;
                                         50               36  C0 000BB        ADDL2   #54, R0                                           ;
            50        08   A8            10               00  ED 000BE        CMPZV   #0, #16, 8(R8), R0                                ; 0732
                                                          27  1E 000C4        BGEQU   10$                                               ;
                                         50        00     BE  3C 000C6        MOVZWL  @0(SP), R0                                        ; 0735
                                         50        2C     A8  C0 000CA        ADDL2   44(R8), R0                                        ;
                                    0C   AC               50  D1 000CE        CMPL    R0, DESIRED_VBN                                   ;
                                                          25  1A 000D2 9$:    BGTRU   12$                                               ;
                                                          69  B7 000D4        DECW    (R9)                                              ; 0738
                                         50        00     BE  3C 000D6        MOVZWL  @0(SP), R0                                        ; 0739
                                    2C   A8               50  C0 000DA        ADDL2   R0, 44(R8)                                        ;
                                         50               69  3C 000DE        MOVZWL  (R9), R0                                          ; 0740
                                         50               06  C4 000E1        MULL2   #6, R0                                            ;
                      00   BE       36   A8               50  28 000E4        MOVC3   R0, 54(R8), @0(SP)                                ;
                                         56               06  C2 000EA        SUBL2   #6, W_POINTER                                     ; 0741
                                         86               5B  B0 000ED 10$:   MOVW    COUNT, (W_POINTER)+                               ; 0747
                                         86               5A  D0 000F0        MOVL    LBN, (W_POINTER)+                                 ; 0748
                                                          69  B6 000F3        INCW    (R9)                                              ; 0750
                                         86        04     AE  F5 000F5 11$:   SOBGTR  COUNTER, 7$                                       ; 0707
                                         50               01  D0 000F9 12$:   MOVL    #1, R0                                            ; 0755
                                                              04 000FC        RET                                                       ; 0757

; Routine Size:  253 bytes,    Routine Base:  $CODE$ + 0000


;   228          0758  1 END
;   229          0759  0 ELUDOM




;                                       PSECT SUMMARY
;
;       Name                          Bytes                         Attributes
;
;  $CODE$                                253  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                Library Statistics
;
;                                              -------- Symbols --------        Pages       Processing
;       File                                   Total    Loaded   Percent        Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1               18619        24         0        1000          00:02.0
;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MWTUR1/OBJ=OBJ$:MWTUR1 MSRC$:MWTUR1/UPDATE=(ENH$:MWTUR1)

; Size:          253 code + 0 data bytes
; Run Time:         00:14.7
; Elapsed Time:     00:34.9
; Lines/CPU Min:     3102
; Lexemes/CPU-Min:  30085
; Memory Used:   153 pages
; Compilation Complete
```

0246 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY